

「ソフトウェア工学」

(1) ドメイン分析・要求分析

東大理学部情報科学科講義

石川 冬樹（本務先：国立情報学研究所）

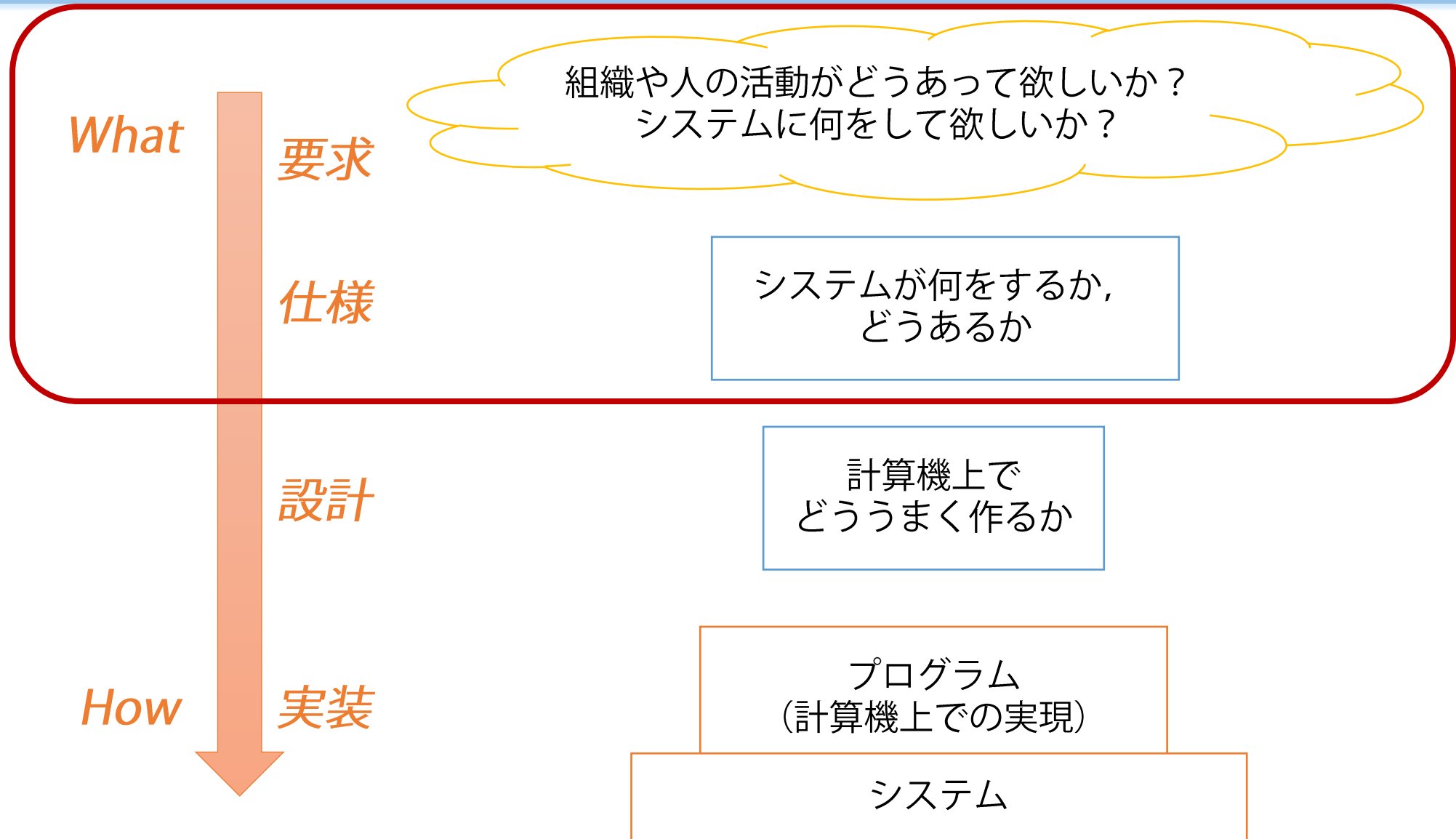
f-ishikawa@nii.ac.jp / @fyufyu

<http://research.nii.ac.jp/~f-ishikawa/>

目次

- ドメイン分析・要求分析概観
- UMLによるモデリング・分析
- ゴール指向要求分析
- ソフトウェア品質

(再掲) ソフトウェア開発における抽象度



ソフトウェア要求と要求定義工程

- **Software Requirement (ソフトウェア要求)** :
実世界の問題を解決するために,
(これから作る) システムにより満たされるべき性質
- 扱う工程は「要求定義」「要件定義」などと呼ばれる
 - 学問としては, **Requirements Engineering (要求工学)**
 - ソフトウェア要求の獲得, 分析, 仕様化, 妥当性確認, 管理
 - **Problem Domain (問題領域)** の理解・分析が重要:
Solution (解決領域) との対比
 - **Stakeholder (利害関係者)** の洗い出し・巻き込みが必要不可欠

要求：重要性

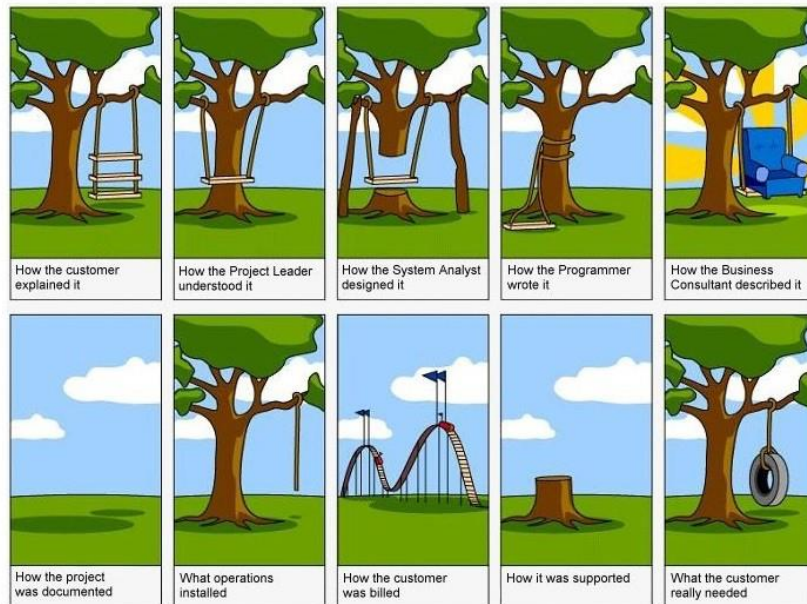
- 要求に問題があると、全開発費の30～50%を消費

[B.W. Boehm et al., Understanding and controlling software costs, 1988]

- 工期遅延理由の55%が要件定義の問題

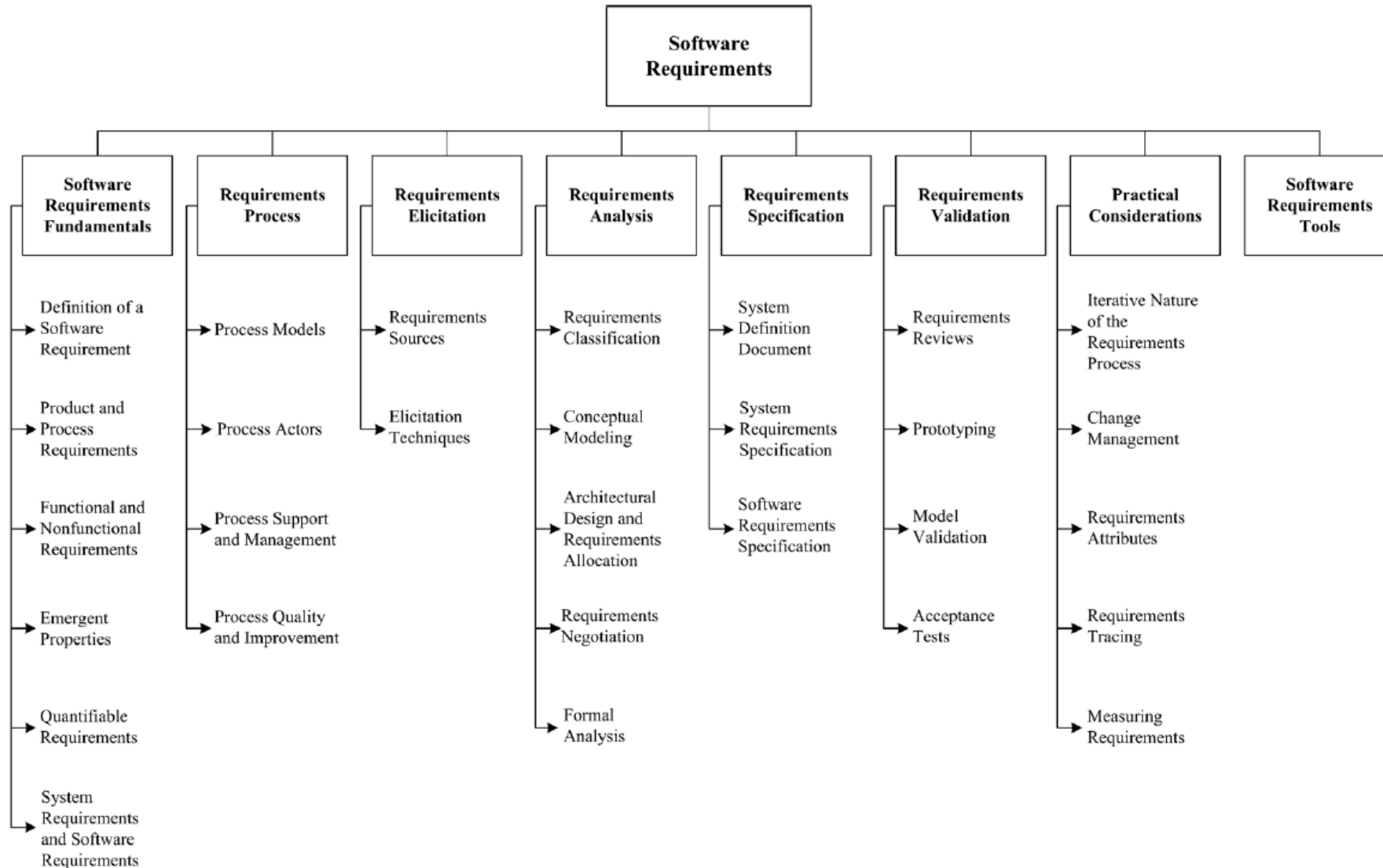
[JUAS・ユーザー企業ソフトウェアメトリックス調査2016]

- 有名な風刺画



[<https://danielksmith.wordpress.com/2012/09/25/what-the-customer-really-wanted/>]
(真の起源はわからず・・・)

要求：関連する知識 (SWEBOKより)



[SWEBOK V3.0]

要求：要求定義工程における活動

■ 要求獲得・抽出 (Elicitation)

- 解くべき問題の理解を通して要求を獲得する

(すでにあるものを) 「抽出」よりも「発見」や「開発」といった語を好む人もいる

■ 要求分析 (Analysis)

- 要求の分類, 競合の検出・解決や, 境界の明確化などの分析を行う

■ 仕様化 (Specification)

- 共有・確認・合意できる文書を構築する

■ 妥当性確認 (Verification and Validation)

- 正しく理解でき, 整合性があり, 完全であることなどを検証し, その妥当性を確認する

要求：満たすべき性質IEEE 830

■ IEEE 830 (1998): IEEE Recommended Practice for Software Requirements Specifications

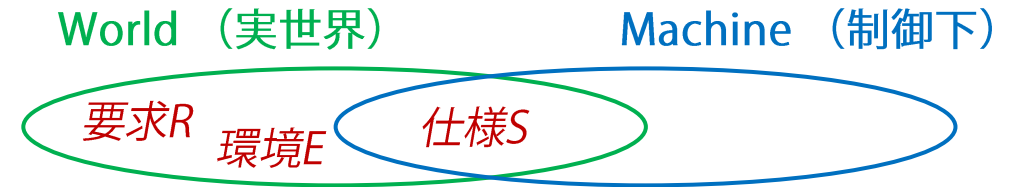
■ 特に下記の品質特性に注目

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

要求：要求の構造と用語（一例）

■ Zave/Jacksonのモデル

- 要求：実世界で達成したいこと
- 仕様：システムが満たすべきこと
- 仮定：環境が満たすべきこと



[画像：<http://www.fujitaka.com/>]

要求R	<ul style="list-style-type: none">enter 発生回数 \leq pay 発生回数
仕様S	<ul style="list-style-type: none">pay の発生を検知したら, unlock するpush の発生を検知したら, lock する
環境E	<ul style="list-style-type: none">push と enter は交互にしか起きないと仮定lock が起きてから unlock が起きるまでは push は起こせないと仮定

$S, E \models R$

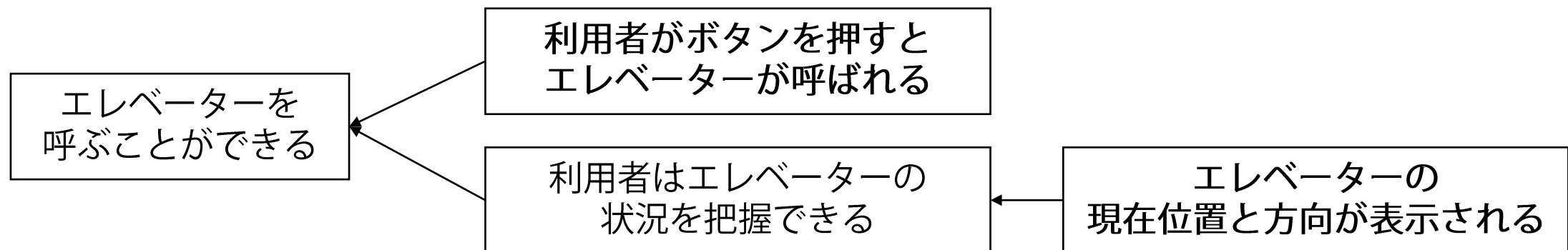
環境Eのもとでシステムにより仕様Sが満たされたとき要求Rが満たされるか？

[Zave et al., Four Dark Corners of Requirements Engineering, 1997]

要求：要求の構造と用語（一例）

■ゴールモデル

- ゴール：（様々なステークホルダーが）達成を望むこと
- サブゴール：あるゴールに対し，それを実現するために必要なゴール（複数必要であったり，選択肢があったりする）
- 要求：ゴールの中で，実現可能性が判断できるレベルに具体化され，かつ（選択肢がある中で）実現すると決めたもの



要求：要求の構造と用語（一例）

■要求仕様

- 要求が陽に文書化された場合に，こう呼ぶことが多い

■実際にはいろいろな用語が使われている

- 標準により異なる

- 組織内用語が産まれる

- 「真の要求」：前頁の用法で言うなら「顕在化しづらいゴール」

- 「要求と要件」：前頁「ゴールと要求」の関係を指すことが多い

- 単に言葉が適当なだけなこと

目次

- ドメイン分析・要求分析概観
- UMLによるモデリング・分析
 - ドメインモデリング・分析
 - 要求モデリング・分析
- ゴール指向要求分析
- ソフトウェア品質

ドメインモデリング・分析

■ Domain Modeling and Analysis

(ドメインモデリング・分析)

- Conceptual Modeling and Analysis (概念…) と呼ぶことも
 - 実世界の問題領域 (ドメイン) において構成要素となる概念やその関係について理解, 表現する
-
- ※ 基本的には現状 (**As-Is** と呼ぶ) を描く
 - これから作るシステムの話 (**ToBe** と呼ぶ) はしていない

ドメインモデリング・分析：デモ

- 航空券予約業務に関する概念のモデリング
- astah* ツール利用

[<https://astah.change-vision.com/ja/index.html>]

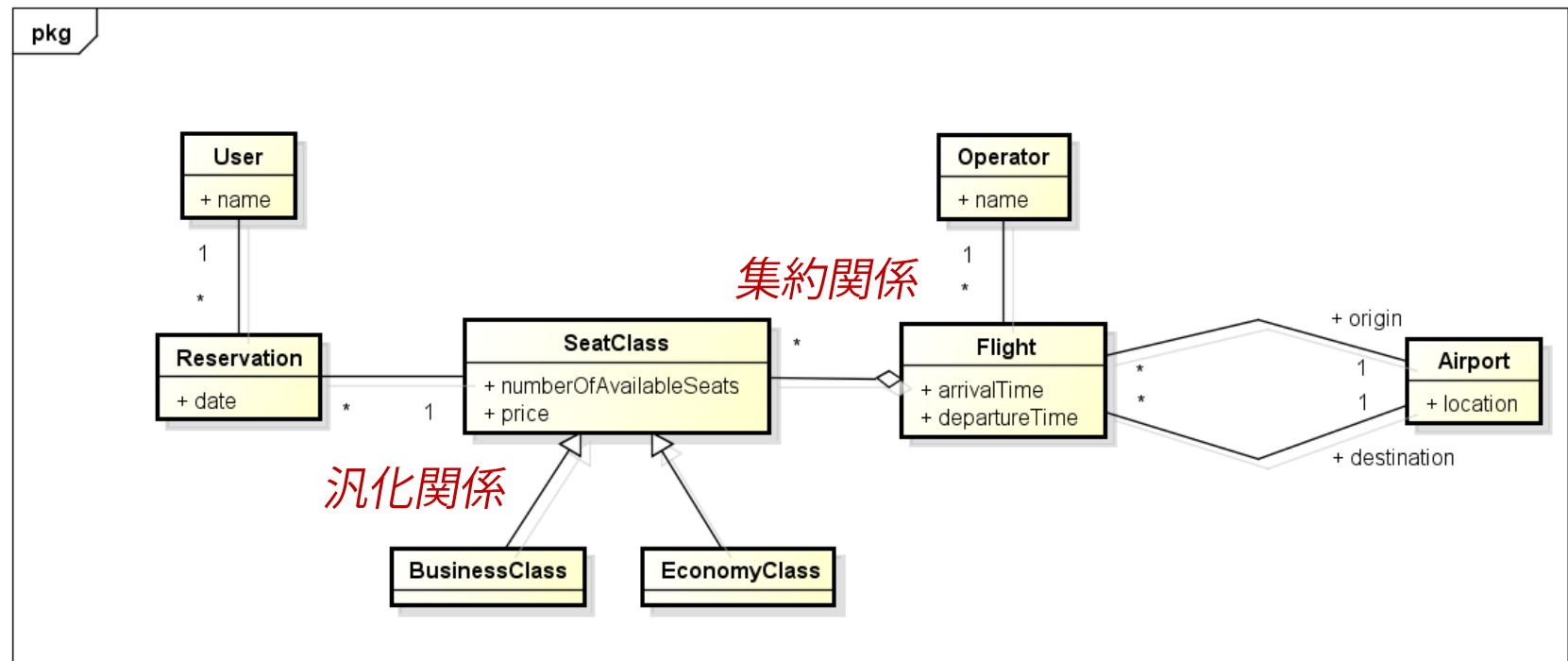
ドメインモデリング：概念と関係の定義記述

■ 航空券予約業務に関する概念（クラス図）

■ 関係

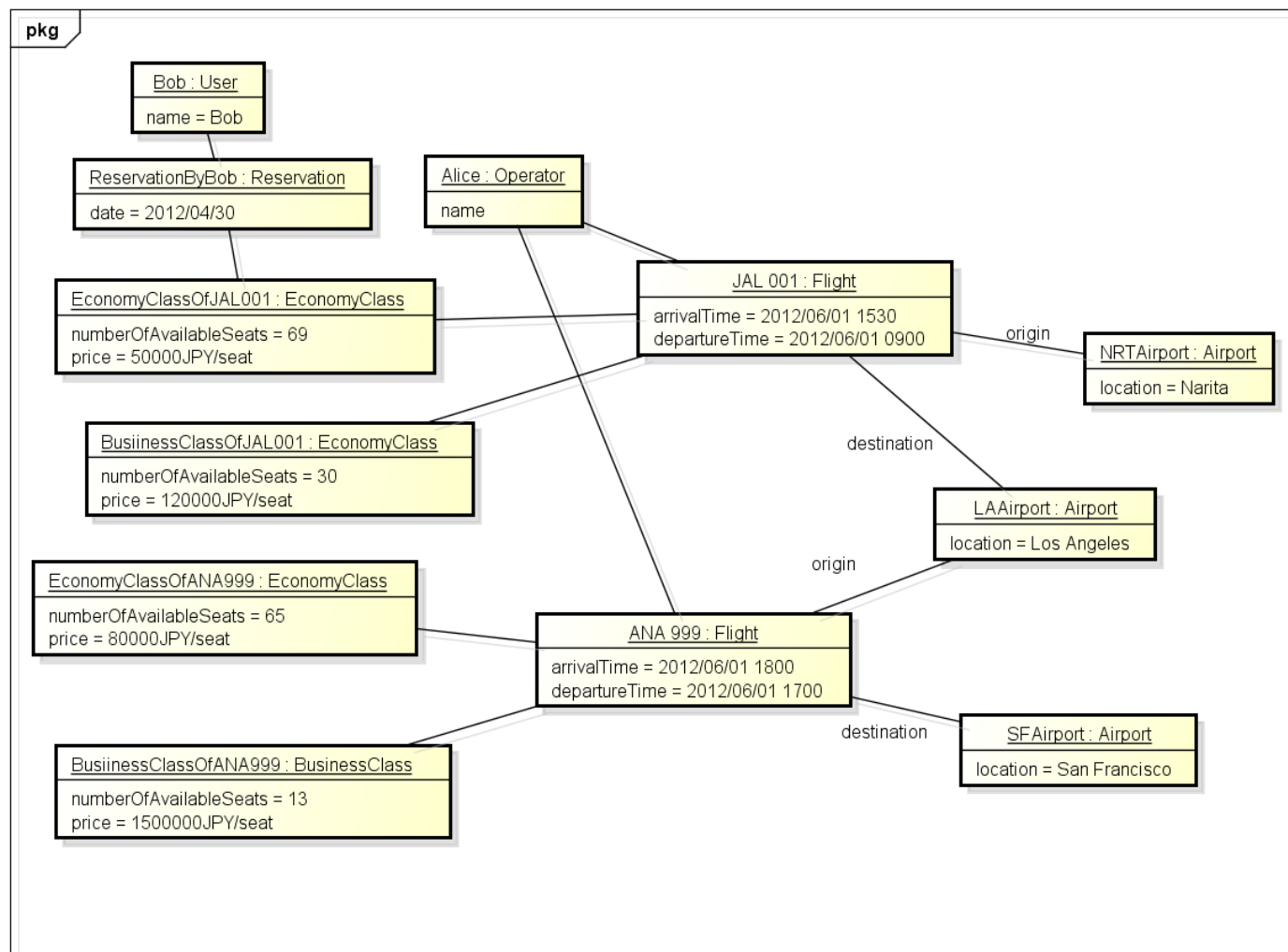
■ 特に，集約関係（has-a関係）・汎化関係（is-a関係）

■ 多重度



ドメインモデリング：概念と関係の具体例記述

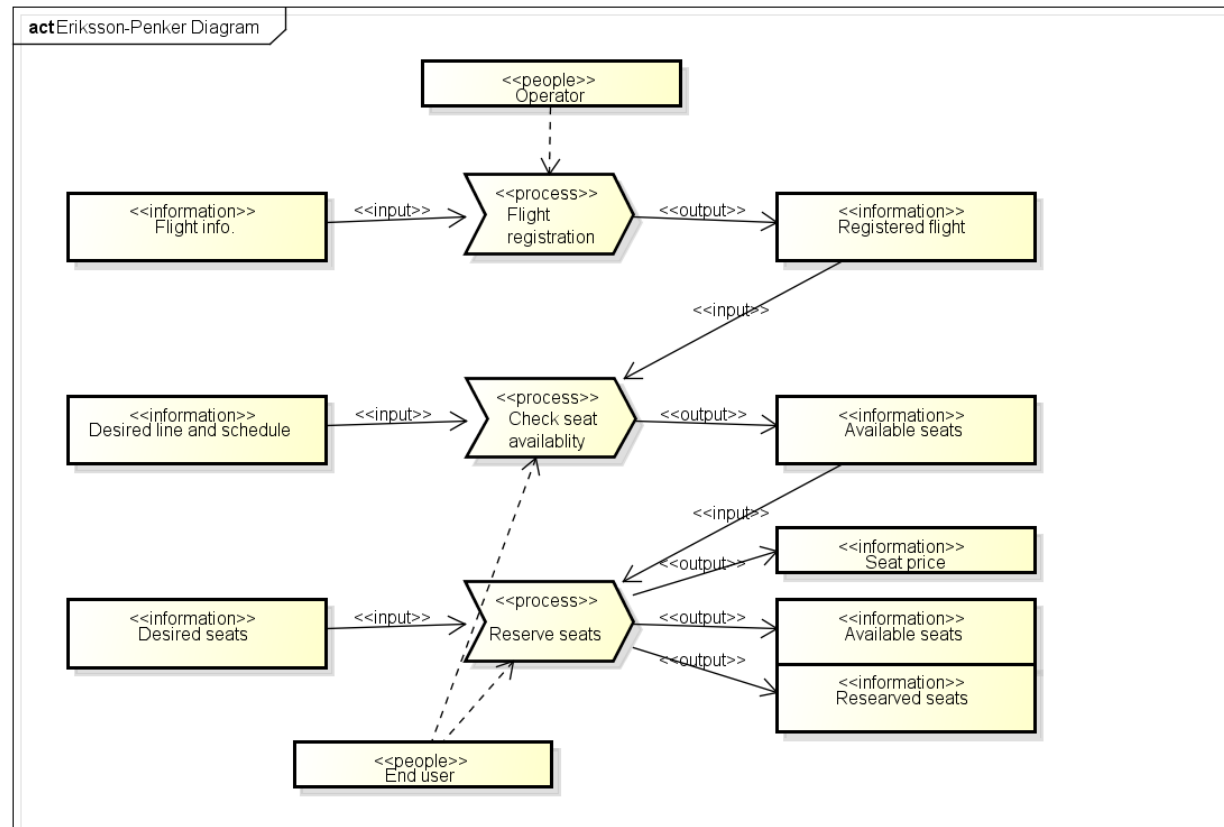
■ 航空券予約業務に関する概念の具体例（インスタンス図）



ドメインモデリング：プロセスの記述

■ 航空券予約業務に関するプロセス

(UML標準の図を拡張した Eriksson-Penker プロセス図)



目次

- ドメイン分析・要求分析概観
- UMLによるモデリング・分析
 - ドメインモデリング・分析
 - 要求モデリング・分析
- ゴール指向要求分析
- ソフトウェア品質

要求モデリング・分析

- 計算機システムの導入による問題解決を考える
 - AsIs を基にして ToBe を描く
- Use Case (ユースケース) :
(これから作る) システムの存在により,
どうユーザタスクが遂行されるかのシナリオを定義

要求モデリング・分析：ヒアリング結果の例

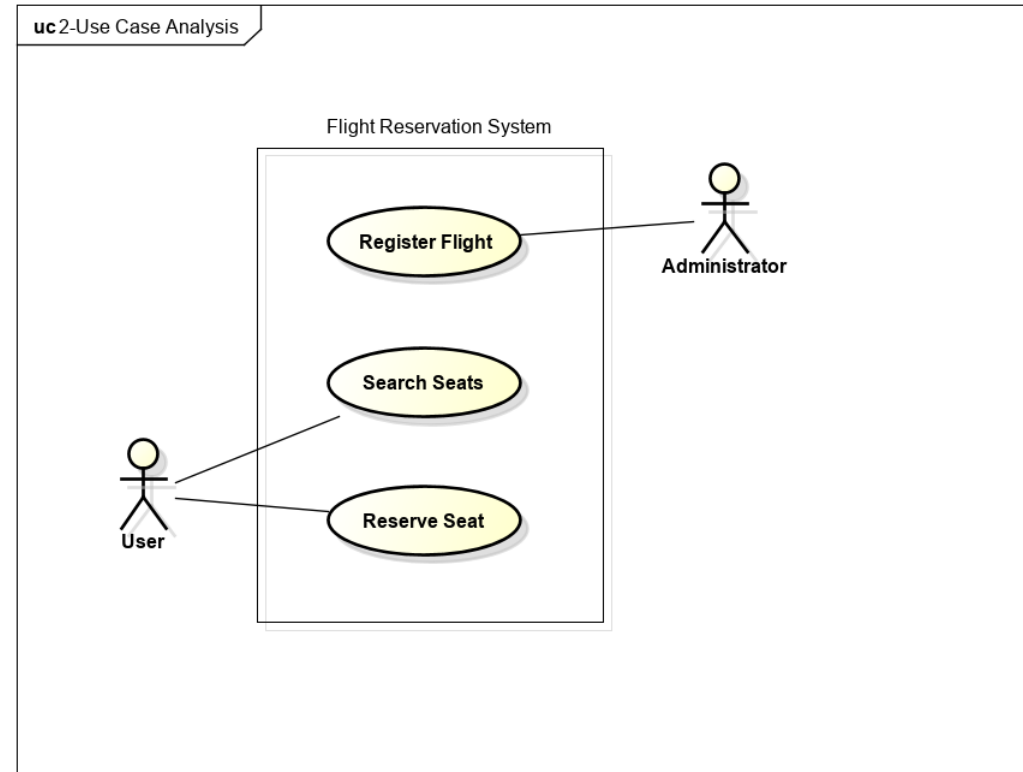
- 航空券予約業務に関するヒアリング結果
 - 昼間に支店を訪れるか電話しないと予約できない
 - ▶ 搭乗ユーザが自身でいつでもどこでも、フライトを検索し予約できるようにしたい
 - 現状の予約状況を人手で整理・把握し、価格に関する意思決定などに活用するのが大変
 - ▶ 自動的に予約状況を整理し、航空会社スタッフが簡単に予約状況を把握できるようにしたい

※ ものすごい昔の例だが…

要求モデリング：ユースケースの概要記述

■ 航空券予約システムにおけるユースケース図

- アクター
- ユースケース
- システム境界



要求モデリング：ユースケースの詳細記述

■ ユースケース記述

Use Case: Register Flight

Actor: Administrator

Purpose: make a new flight accessible on the system

Precondition: none

Postcondition: a flight with inputted route and schedule is registered

Basic Sequence:

1. The actor indicates an intention to register a new flight
2. The system prompts the actor to input the route (departure, destination) and schedule (departure, arrival), as well as price and the number of available seats for each class
3. The actor inputs the information to the system
4. The system registers the flight information
5. The system shows the registered information to the actor

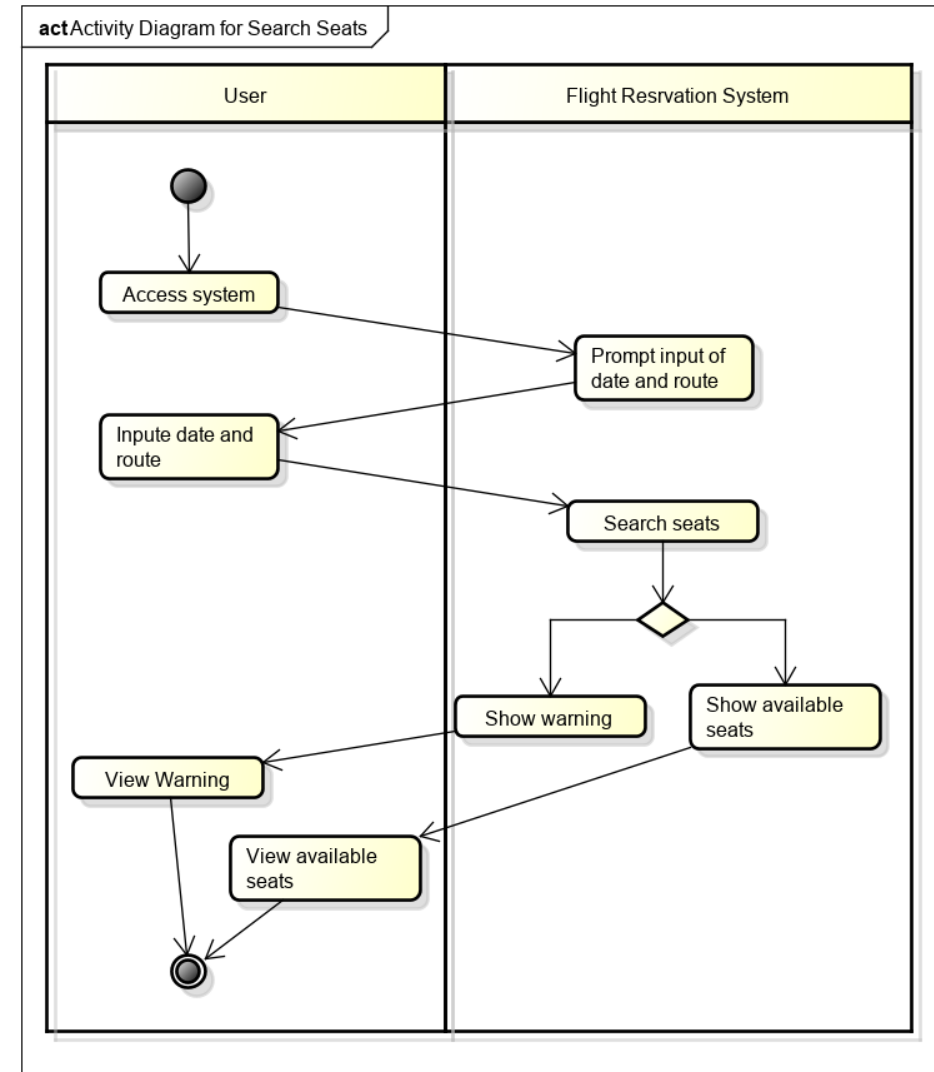
Alternative Sequence: none

Remarks: none

Scenario: Bob with the administrator role indicates ... inputs a flight information from Tokyo Narita to Paris CDG ...

要求モデリング：ユースケースにおけるフロー記述

- ユースケースにおけるシーケンスを表したアクティビティ図
 - アクターとシステム (パーティション)
 - 順序・分岐



目次

- ドメイン分析・要求分析概観
- UMLによるモデリング・分析
- ゴール指向要求分析
- ソフトウェア品質

ゴール指向要求分析

■ Goal-Oriented Requirements Analysis

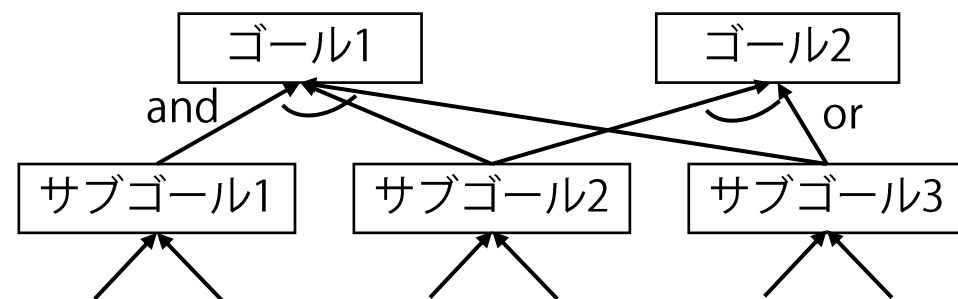
(ゴール指向要求分析)

- UMLはユースケース記述から出発するが、それを得る段階が重要
- ➡ ゴール（達成や保証をしたいことから）を分析すべき
- ゴールの完全性や合目的性の検証が重要
- ➡ 「○○のために××が必要」という関係性の階層構造の明示化
 - 目的に対する手段の完全性や、手段の合目的性の検証
 - 代替手段の追求
 - 変更発生時における影響範囲の把握

ゴールモデル：一般的な概念

■ AND/OR decomposition (分解・詳細化) の多段階構造

- 一般にはDAG：一つのサブゴールは複数の上位ゴールに関連しうる, 関連のループしない



■ ハードゴールとソフトゴールの区別

- ゴールの達成がtrue/falseで判断されるか, 数値指標など度合いで判断されるか
- 性能やコストなど**非機能要求**はソフトゴールになることが多い

ゴールモデル：Why と How の行き来

■ ゴールモデルの構築手順

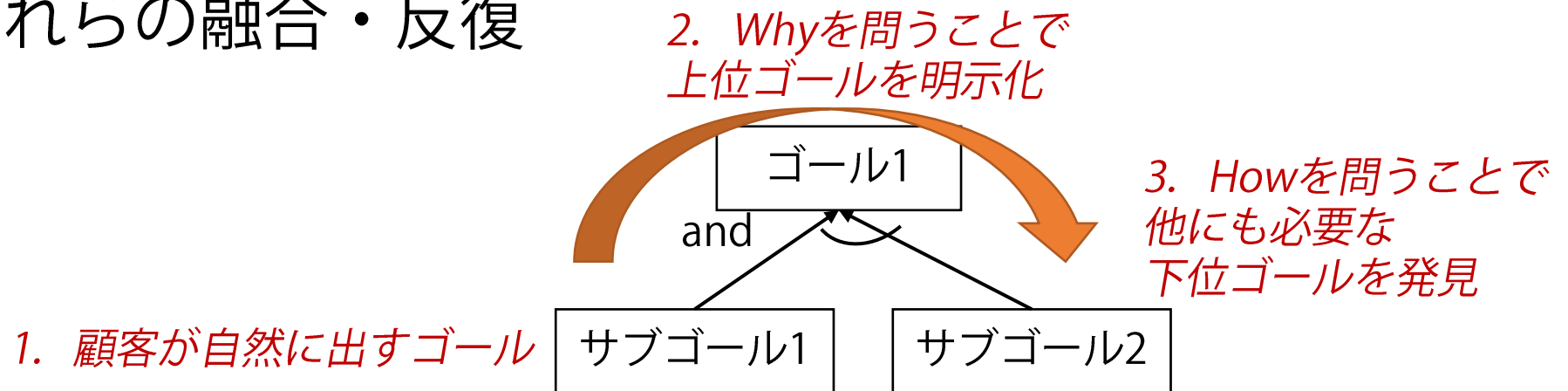
■ 「上（抽象）から下（具体）へ」

→ How（どのように実現していくか？）を問うてゆく

■ 「下（具体）から上（抽象）へ」

→ Why（なぜそのゴールが必要か？）を問うてゆく

■ 実際はこれらの融合・反復

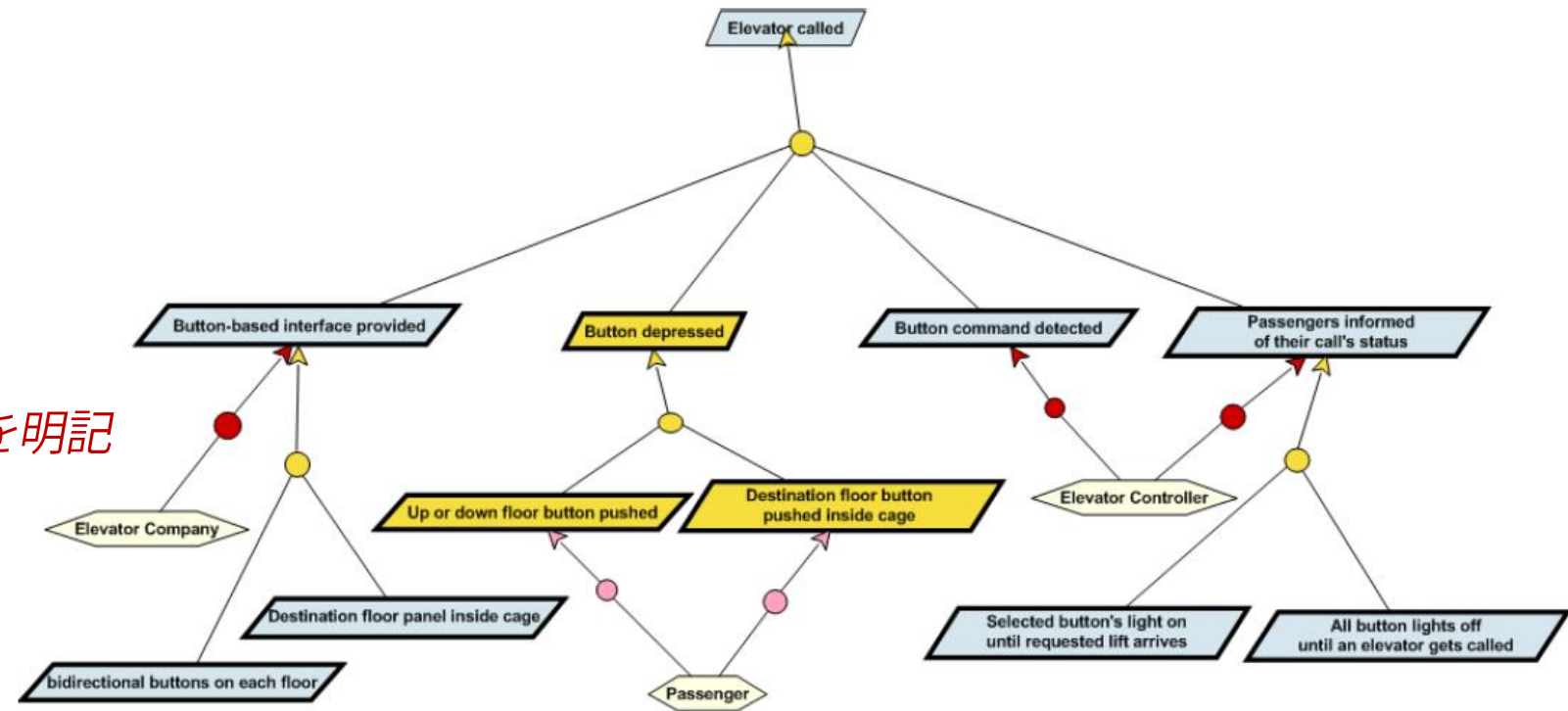


ゴール指向要求分析の手法例：KAOS

- エージェント（部品や外界要素）への責務割り当てまでを行う枠組み

エージェントへの責務割当を明記

末端のゴールが「(システムへの) 要求」



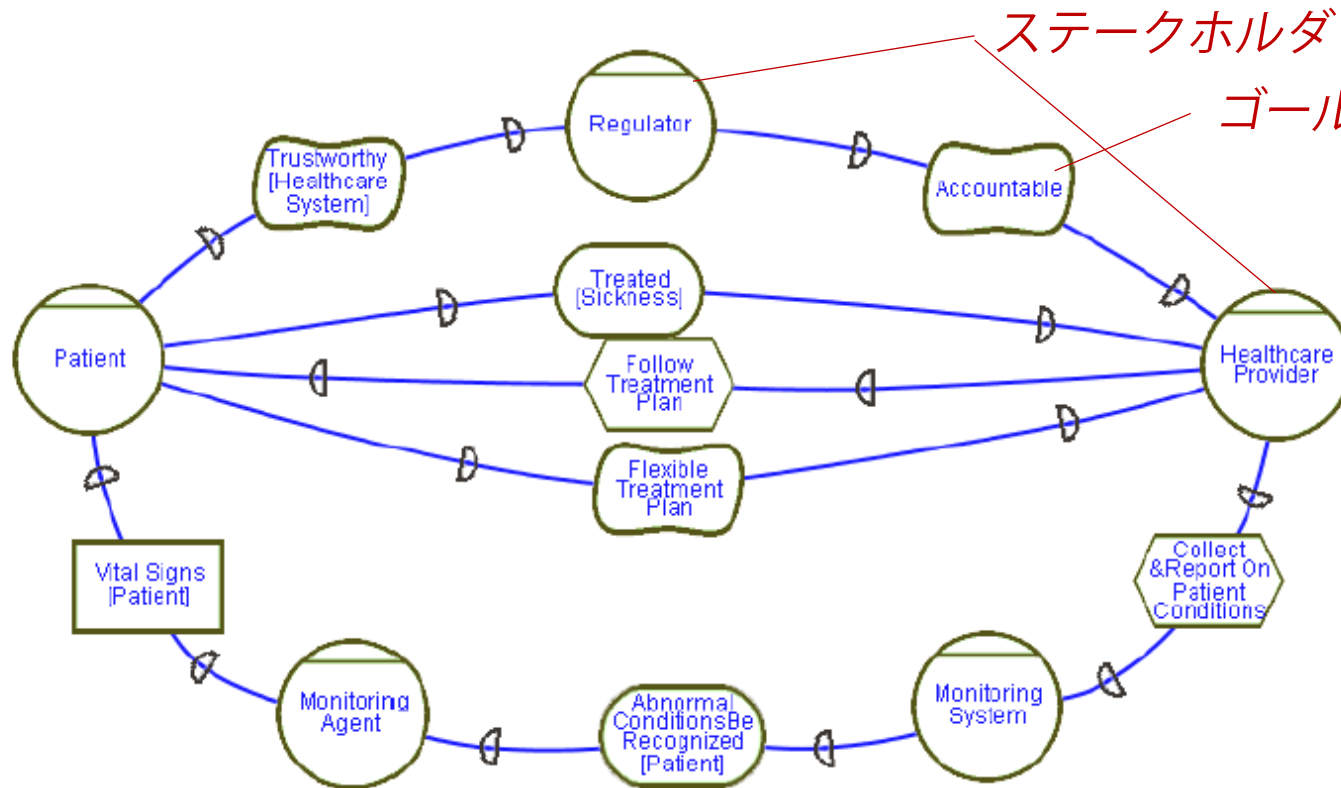
利用者に割り当てられたゴール（黄色）は「仮定・期待」を表す

[<http://www.objectiver.com/>] より

ゴール指向要求分析の手法例：i*（アイスター）

■ 複数ステークホルダーの関係性分析を重視する手法

■ 下はAsIsのSDモデル（Strategic Dependency）の例



ステークホルダ

ゴールを通した依存関係

どういう依存しているか・頼っているか

エッジにおいて

- D記号の手前側が依存している側
- D記号の指す先が依存されている側
- エッジ内には関連するゴールノード

[E. Yu, Social Modeling and i*] より

ゴール指向要求分析の手法例：i*（アイスター）

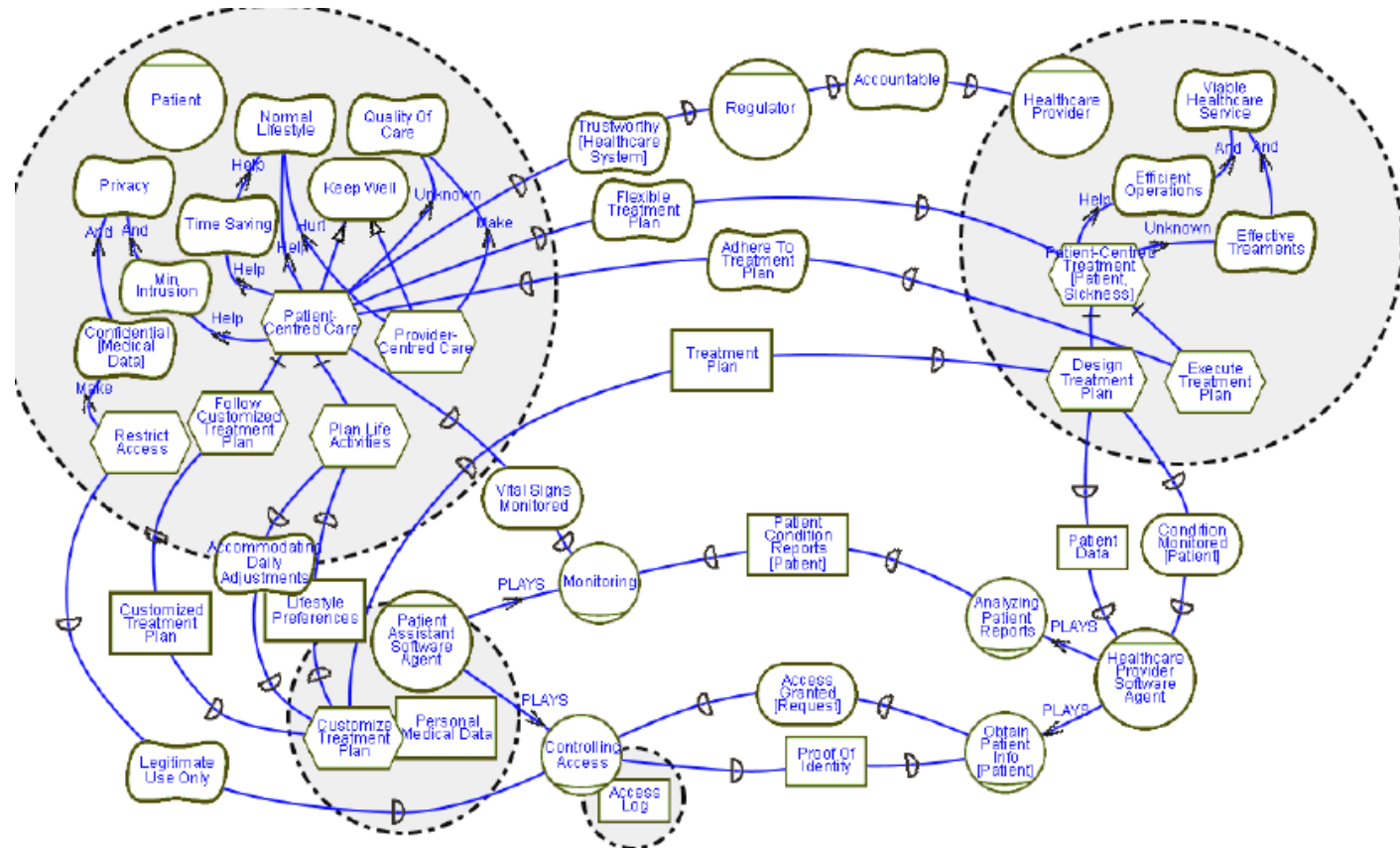
■（続）

■下はToBeのSRモデル（Strategic Rationale）の例

前頁の図から、
ステークホルダ内部の
ゴールモデルを明確化

サブゴールの
上記ゴールへの
寄与度が区別
されている

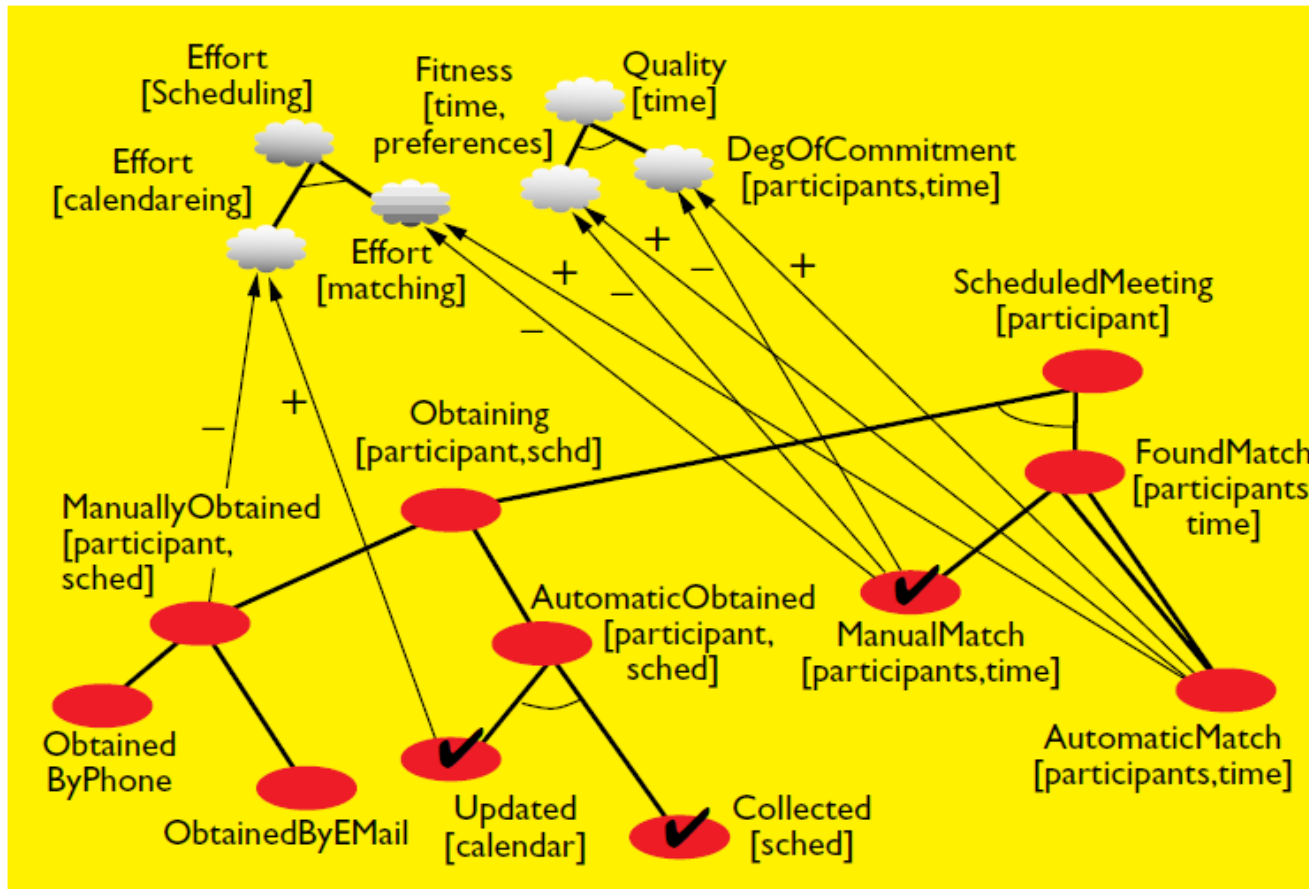
Make, Help,
Unknownなど



[E. Yu, Social Modeling and i*] より

ゴール指向要求分析の手法例：NFR

■ 非機能要求 (Non-Functional Requirements) を重視



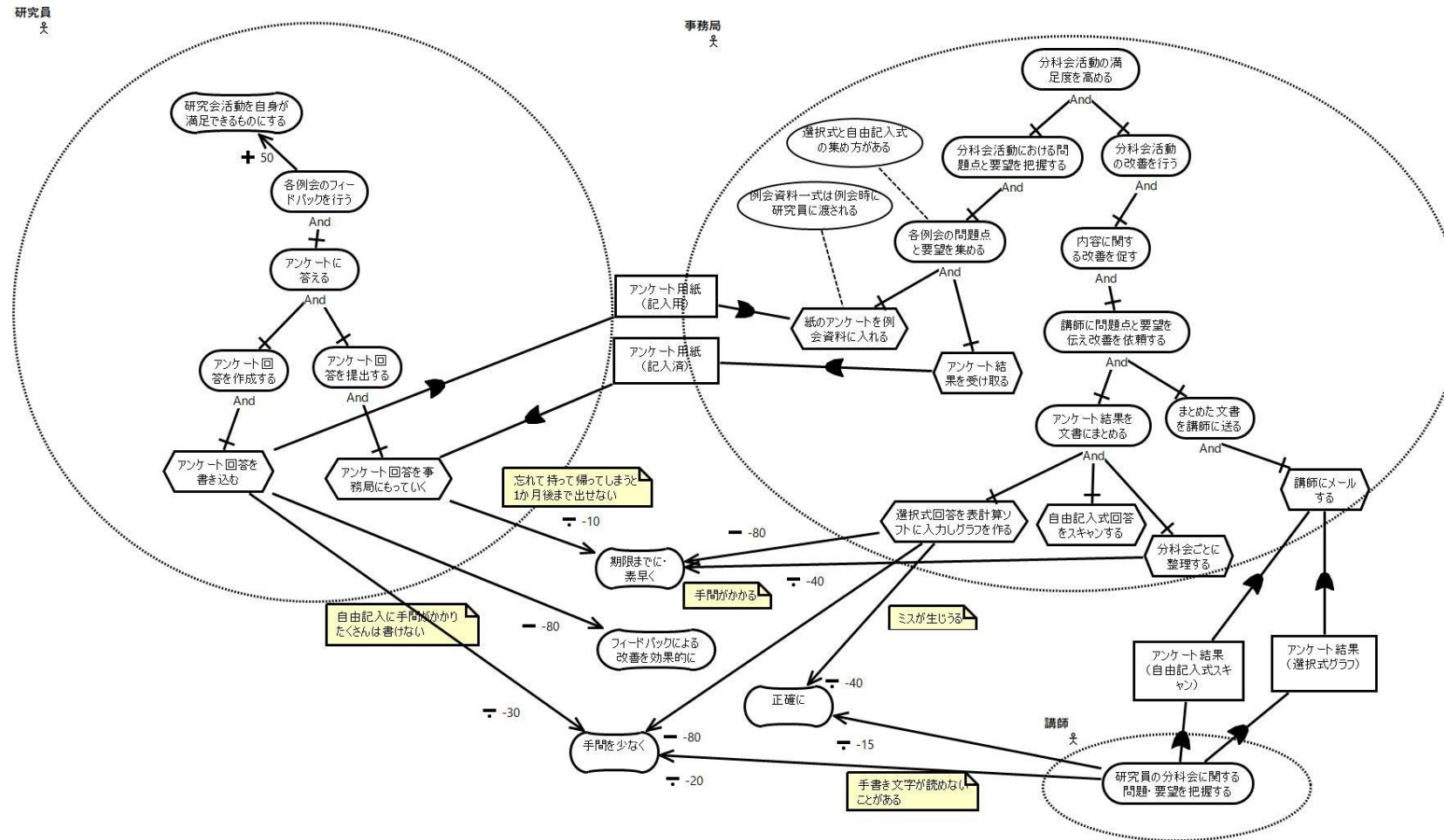
非機能要求を表すソフトゴール（雲形）とそれに対するプラスマイナスの影響

or分解で示された
代替えゴール
（実現手段の選択肢）
について非機能要求の
観点から評価

[J. Mylopoulos et al., From object-oriented to goal-oriented requirements analysis, CACM'99]

ゴール指向要求分析の記述例 (1)

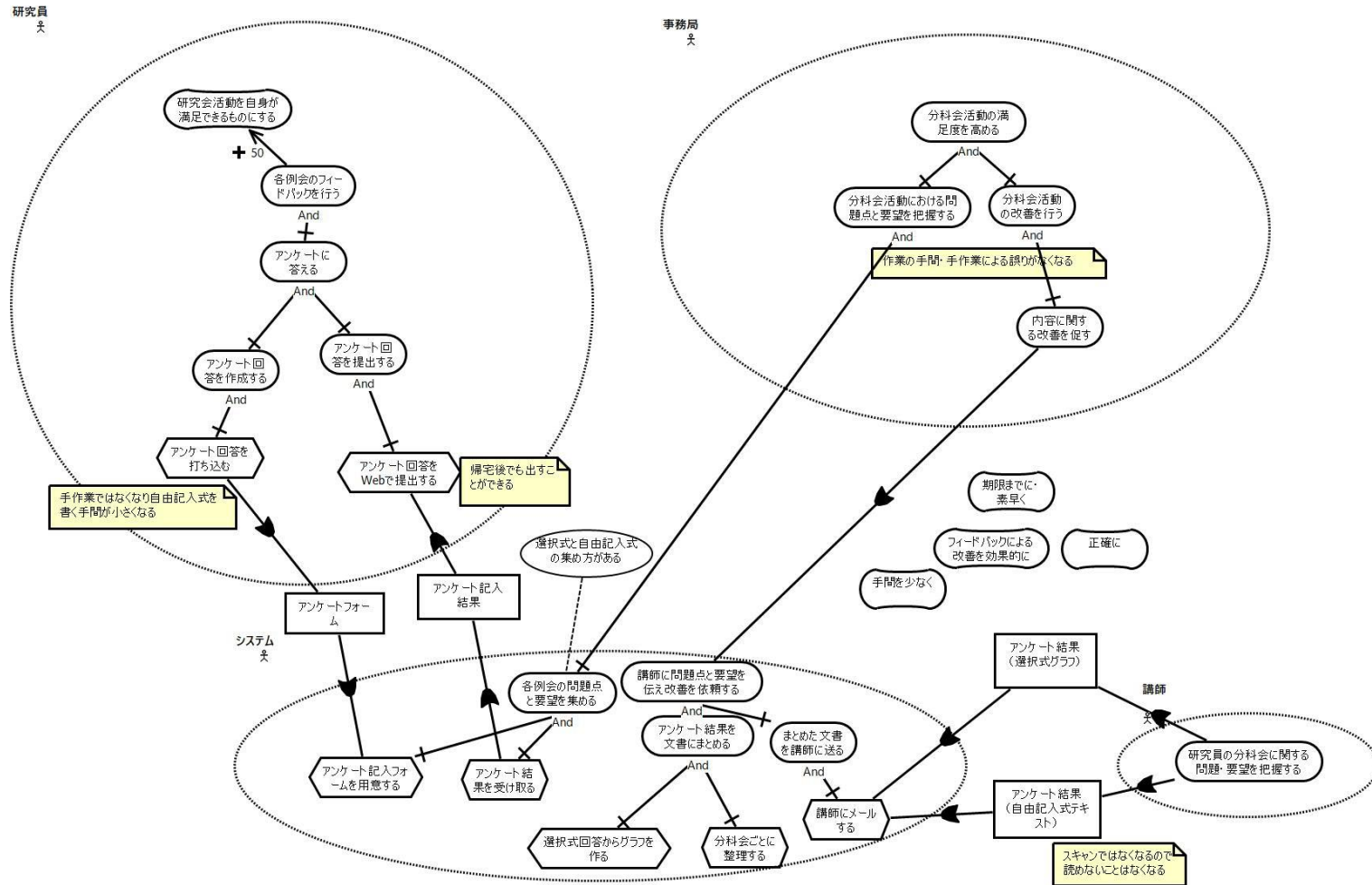
■ 企業向けセミナーにおける受講アンケートに関する AsIs



User Requirements Notation (URN) ($i^* + NFR$) 記法による

ゴール指向要求分析の記述例 (2)

■企業向けセミナーにおける受講アンケートに関する ToBe



User Requirements Notation (URN) ($i^* + NFR$) 記法による

ゴール指向要求分析の実際

- 以上の手法例は初期の有名なもの
 - これらを実務でそのまま使っている状況はおそらく限られる
- 「〇〇のために××が必要」という関係性に着目した構造化, 分析は非常に普遍的で, 実務でも重視される
 - スプレッドシート上で2-3段階の深さで行うことは一般的

売上分析 (URIAGE)	要求	URIAGE-01	画面のドロップダウンリストで店舗を指定するところから、店舗コードを取得し、取得した店舗コードをもとに売上テーブルを検索し、検索した結果を編集し、編集したデータをもとに売上一覧表を表示するところまで行う。
		理由	店舗ごとの売上の傾向を知りたい、それで売上が少ない店舗を対策を早め立てたい、売れ筋の商品を知りたい。
		説明	実装時期は、10月ごろ。
			<ドロップダウンリストの表示>
		□□□ URIAGE-01.001	店舗テーブルを次の条件で検索し、店舗コードと店舗名を取得する。 <ul style="list-style-type: none">・廃止した店舗（廃止フラグが1のもの）は除く・有効期限内（本日が、開店日と終了日の間にある）・店舗コードの昇順
		□□□ URIAGE-01.002	取得した店舗コードは、ドロップダウンリストの値として設定する。
		□□□ URIAGE-01.003	取得した店舗名は、ドロップダウンリストの表示名として設定する。
			<ドロップダウンリストの指定後>
		□□□ URIAGE-01.101	ドロップダウンリストの指定が発生した場合（changeイベントが発生した場合）に、売上一覧表処理を実行するようにする。

USDMという記述指針の例
[https://affordd.jp/previous/tech_documents/affordd-t2-usdmttext-basic_1.3.pdf] より引用

補足：要求工学におけるその他の話題

- ビジネス分析のための手法
 - SWOT分析, ビジネスモデルキャンパスなど
- アイディア創出・整理のための手法
 - 「超上流要求工学」とも呼ばれる
 - ブレインストーミング, KJ法など
- 要求獲得・コミュニケーションのための手法
 - インタビュー, エスノグラフィ, ペルソナなど
- 要求妥当性確認の手法
 - プロトタイピングなど

目次

- ドメイン分析・要求分析概観
- UMLによるモデリング・分析
- ゴール指向要求分析
- ソフトウェア品質

品質の概念

■ 「品質」という概念の定義例

対象に本来備わっている特性の集まりが、
要求事項を満たす程度

(ISO 9000・品質マネジメントシステムに関する標準)

様々なステークホルダーの明示的または暗黙的な
ニーズを満たす、すなわち価値を提供する程度

(ISO/IEC 25010・ソフトウェア製品の品質要求および評価に関する標準)

- 様々な特性（側面・観点）から構成
- ニーズや要求事項に対して相対的
- 多くの非機能要求を含む

品質に関する意思決定の構造

枠組み・ガイドライン

システムの重要さや 要求の強さの区分

- 企業内利用なら
「社会的影響が殆どないシステム」
- ...
- 国民生活・社会経済のインフラなら
「社会的影響が極めて大きいシステム」

品質特性
(観点)

具体化

評価指標

- 可用性 → 稼働率, 障害復旧水準, ...
- 性能 → 時間性能 → 応答時間, ...
→ 空間性能 → 最大受容データ量, ...
- ...

求める
基準

かけ合わせて設定

- 「社会的影響が極めて大きい」ならば,
- 「稼働率」は「1年間で数分程度の停止まで許容」
 - 「応答時間」のサービスレベルを規定すること
 - ...

品質に関する意思決定の構造

枠組み・ガイドライン

システムの重要さや 要求の強さの区分

- 企業内利用なら
「社会的影響が殆どないシステム」
- ...
- 国民生活・社会経済のインフラなら
「社会的影響が極めて大きいシステム」

品質特性
(観点)

具体化

評価指標

- 可用性 → 稼働率, 障害復旧水準, ...
- 性能 → 時間性能 → 応答時間, ...
→ 空間性能 → 最大受容データ量, ...
- ...

求める
基準

かけ合わせて設定

- 「社会的影響が極めて大きい」ならば,
- 「稼働率」は「1年間で数分程度の停止まで許容」
 - 「応答時間」のサービスレベルを規定すること
 - ...

個別の適用

今回は...

「社会的影響が極めて大きいシステム」

ここが契約に反映

- 「稼働率」は
「1年間で数分程度の停止まで許容」
- 「応答時間」は
「平均が3秒以内」と規定
- ...

SQuaRE (ISO 250XXシリーズ) : 概要

■ ソフトウェアシステムに関する品質モデルを定義

■ SQuaRE: Systems and software Quality Requirements and Evaluation

■ 品質モデル :

どういう観点で品質要求を定め評価するのかという
品質特性 (の階層構造) を定義

例 :

プロダクト品質

- 信頼性

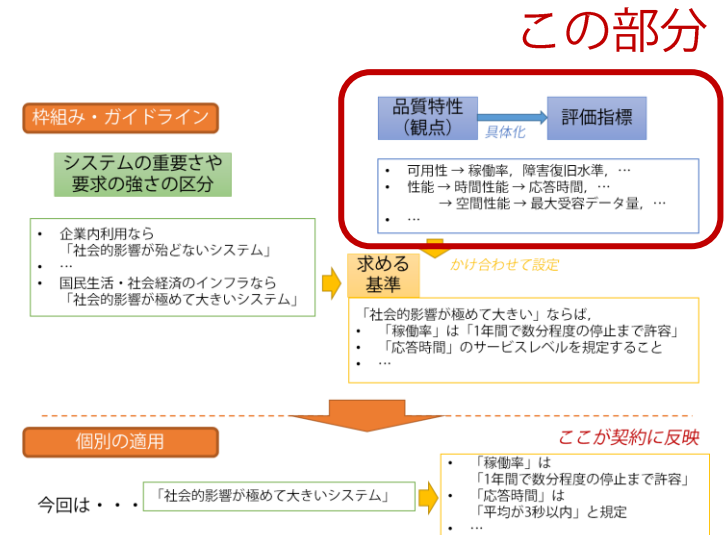
-- 成熟性

-- 平均故障間隔

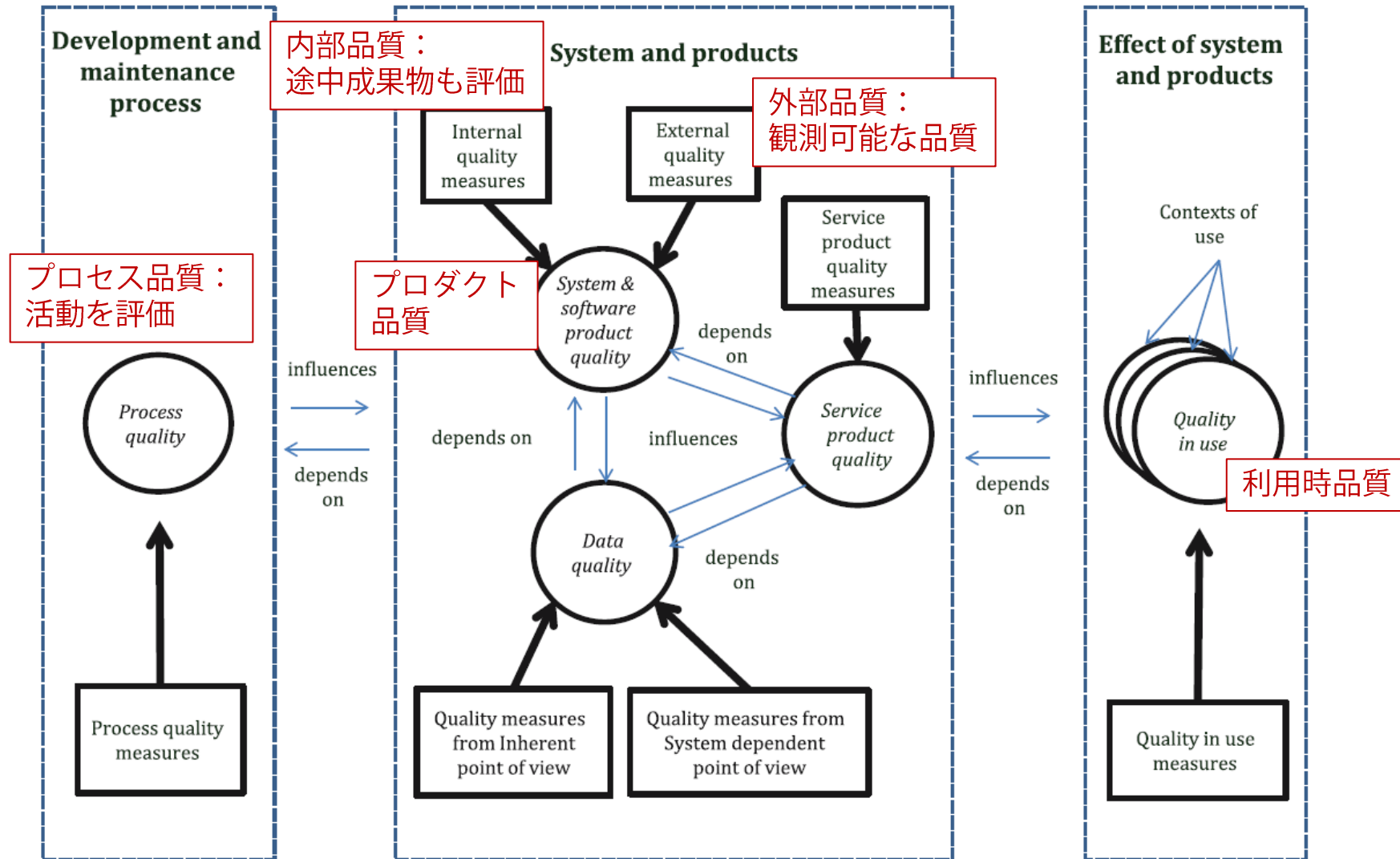
品質特性

品質副特性

品質測定量要素



SQuaRE (ISO 250XXシリーズ) : 概要品質特性の分類



SQuaRE (ISO 250XXシリーズ) : プロダクト品質特性

■ SQuaRE Software Product Quality (ISO/IEC 25010)

Functional Suitability

- Functional Completeness
- Functional Correctness
- Functional Appropriateness

Performance Efficiency

- Time Behavior
- Resource Utilization
- Capacity

Security

- Confidentiality
- Integrity
- Non-repudiation
- Authenticity
- Accountability

Compatibility

- Co-existence
- Interoperability

Usability

- Appropriateness
- Recognizability
- Learnability
- Operability
- User Error Protection
- User Interface Aesthetics
- Accessibility

Reliability

- Maturity
- Availability
- Fault Tolerance
- Recoverability

Maintainability

- Modularity
- Reusability
- Analyzability
- Modifiability
- Testability

Portability

- Adaptability
- Installability
- Replaceability

SQuaRE (ISO 250XXシリーズ) : 利用時品質特性

■ SQuaRE Quality in use (ISO/IEC 25010)

Satisfaction
- Usefulness
- Trust
- Pleasure
- Comfort

Effectiveness
- Effectiveness

Efficiency
- Efficiency

Freedom from risk
- Economic risk mitigation
- Health and safety risk mitigation
- Environmental risk mitigation

Context Coverage
- Context completeness
- Flexibility

非機能要求グレード：概要

- 「システムの種別 → 非機能要求の強さ」の指針
- IPA, 初期版が2010年, 更新継続

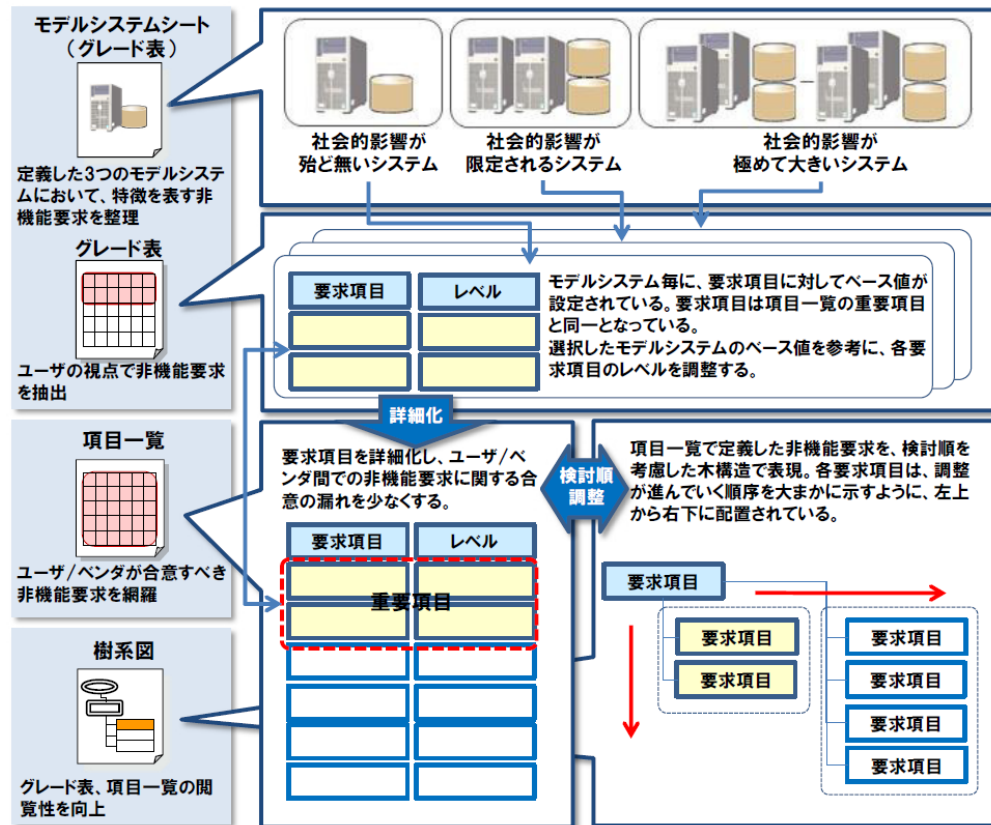


図 1.4.2.1 非機能要求グレードの概要と全体イメージ

[非機能要求グレード2018 利用ガイド解説編より
<https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>]

非機能要求グレード：具体例

■品質特性とそのレベル分け

大項目	中項目	小項目	小項目説明	重複項目	メトリクス (指標)	レベル					
						0	1	2	3	4	5
可用性	継続性	運用スケジュール	システムの稼働時間や停止運用に関する情報。		運用時間(通常)	規定無し	定時内 (9時～17時)	夜間のみ 停止 (9時～21時)	1時間程度の 停止有り (9時～翌 朝8時)	若干の停 止有り (9時～翌 朝8時55分)	24時間無 停止

■システム区分に応じたレベル選択ベース値

社会的影響が殆ど無いシステム		社会的影響が限定されるシステム		社会的影響が極めて大きいシステム	
選択レベル	選択時の条件	選択レベル	選択時の条件	選択レベル	選択時の条件
2	夜間のみ 停止 (9時～21時) 夜間に実施する業務はなく、システムを停止可能。 [-] 運用時間をもっと限って業務を稼働させる場合 [+] 24時間無停止やリポート処理等の短時間の停止のみを考える場合	4	若干の停 止有り (9時～翌 朝8時55分) 24時間無停止での運用は必要ないが、極力システムの稼働は継続させる。 [-] 夜間のアクセスは認めないなど、長時間運用を停止する場合 [+] 24時間無停止で運用する場合	5	24時間無 停止 システムを停止できる時間帯が存在しない。 [-] 1日のスケジュールで定期的に運用を停止する時間帯が存在する場合

[非機能要求グレード2018 グレード表より
<https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>]

重要な点：要求の不確かさ

■対象とするビジネスや人の活動について

「漏れなく」「やるべきこと」をすべて洗い出すのは困難

■もしもそれで800ページの要求仕様書ができたとして、
確信をもって送り出せるか？

■1年かけて作っている間に状況が変わらないか？

➡「今重要だと考えている小さい部分」に絞って取り組み、
小さな単位で開発・評価を反復する手もあるのでは？

➡「アジャイルソフトウェア開発」（第5回に解説）

今回の参考文献 (1)

■ 要求工学知識体系 (REBOK) 概観

- IPAのスライド (2012)

[<https://www.ipa.go.jp/files/000005375.pdf>]

- 分野の概観としてよい

■ 要求工学知識体系 (REBOK)

- 3冊の書籍 (BOKそのもの, 各企業での手法, 事例)

- 近代科学社, 情報サービス産業協会REBOK企画WG

- BOKそのものは用語の羅列なのでリファレンス用

■ ソフトウェア要求

- K. Wiegers et al., 渡部ら訳, 日経BP, 2014

今回の参考文献 (2)

- ゴール指向による!!システム要求管理技法

- 山本 修一郎, ソフトリサーチセンター, 2007

- ゴール指向で日本語の本なら

- 初期の手法を振り返る

- Objectiver

[<http://www.objectiver.com/>]

- ゴール指向のKAOS手法のツール

- チュートリアルなどもあり

- i* Web Site

- i* に関する情報集約

[<http://www.cs.toronto.edu/km/istar/>]

まとめ

■ ドメイン分析・要求分析

- プロジェクトの成否に大きくかかわる最も重要な活動
- 問題領域を理解し, AsIs から ToBe を描き,
これから作るシステムが何をすべきかを定義する
- 様々なステークホルダーがかかわり, 業務や組織, 社会にかかわる難しい (正解がない) 活動に対し,
系統的に取り組み効果・効率を挙げることを目指す